

```

JJJ      000000000      BBBB BBBB BBBB      CCCCCCCCCC      TTTTTTTTTTTTTTTT      LLL
JJJ      00C000000      BBBB BBBB BBBB      CCCCCCCCCC      TTTTTTTTTTTTTTTT      LLL
JJJ      000000000      BBBB BBBB BBBB      CCCCCCCCCC      TTTTTTTTTTTTTTTT      LLL
JJJ      000      000      BBB      BBB      CCC      TTT      LLL
JJJ      000      000      BBB      BBB      CCC      TTT      LLL
JJJ      000      000      BBB      BBB      CCC      TTT      LLL
JJJ      000      000      BBB      BBB      CCC      TTT      LLL
JJJ      000      000      BBB      BBB      CCC      TTT      LLL
JJJ      000      000      BBB      BBB      CCC      TTT      LLL
JJJ      000      000      BBB      BBB      CCC      TTT      LLL
JJJ      000      000      BBBB BBBB BBBB      CCC      TTT      LLL
JJJ      000      000      BBBB BBBB BBBB      CCC      TTT      LLL
JJJ      000      000      BBBB BBBB BBBB      CCC      TTT      LLL
JJJ      000      000      BBB      BBB      CCC      TTT      LLL
JJJ      000      000      BBB      BBB      CCC      TTT      LLL
JJJ      000      000      BBB      BBB      CCC      TTT      LLL
JJJ      000      000      BBB      BBB      CCC      TTT      LLL
JJJ      000      000      BBB      BBB      CCC      TTT      LLL
JJJ      000      000      BBB      BBB      CCC      TTT      LLL
JJJ      000      000      BBB      BBB      CCC      TTT      LLL
JJJJJJJJ  000000000      BBBB BBBB BBBB      CCCCCCCCCC      TTT      LLLLLLLLLLLLLLLLL
JJJJJJJJ  000000000      BBBB BBBB BBBB      CCCCCCCCCC      TTT      LLLLLLLLLLLLLLLLL
JJJJJJJJ  000000000      BBBB BBBB BBBB      CCCCCCCCCC      TTT      LLLLLLLLLLLLLLLLL

```

[illegible][illegible]

RRRRRRRR RRRRRRRR RR RR RR RR RRRRRRRR RRRRRRRR RR RR RR RR RR RR RR	EEEEEEEEEE EEEEEEEEEE EE EE EE EE EEEEEEEEEE EEEEEEEEEE EE EE EE EEEEEEEEEE EEEEEEEEEE	SSSSSSSS SSSSSSSS SS SS SS SS SSSSSS SSSSSS SS SS SS SSSSSS SSSSSSSS	TTTTTTTTTT TTTTTTTTTT TT TT TT TT TT TT TT TT TT TT TT	RRRRRRRR RRRRRRRR RR RR RR RR RRRRRRRR RRRRRRRR RR RR RR RR RR RR RR	IIIIII IIIIII II II II II II II II II II IIIIII IIIIII	CCCCCCCC CCCCCCCC CC CC CC CC CC CC CC CC CC CCCCCCCC CCCCCCCC	TTTTTTTTTT TTTTTTTTTT TT TT TT TT TT TT TT TT TT TT TT TT	
LL LL LL LL LL LL LL LL LL LL LL LLLLLLLLLL LLLLLLLLLL	IIIIII IIIIII II II II II II II II II II IIIIII IIIIII	SSSSSSSS SSSSSSSS SS SS SS SS SSSSSS SSSSSS SS SS SS SSSSSS SSSSSSSS						

I 5
16-Sep-1984 00:24:02
14-Sep-1984 12:37:13VAX-11 Bliss-32 V4.0-742
[JOBCTL.SRC]RESTRICT.B32;1

```
1 0001 0 MODULE RESTRICT (%TITLE 'Restricted login hours enforcement'
2 0002 0 IDENT = 'V04-000'
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 * ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 * TRANSFERRED.
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 * CORPORATION.
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1
31 0031 1 ++
32 0032 1 FACILITY:
33 0033 1 Job controller.
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1 This module contains the routines that enforce restricted login hours.
37 0037 1
38 0038 1 ENVIRONMENT:
39 0039 1 VAX/VMS user and kernel mode.
40 0040 1 --
41 0041 1
42 0042 1 AUTHOR: M. Jack, CREATION DATE: 16-Feb-1982
43 0043 1
44 0044 1 MODIFIED BY:
45 0045 1
46 0046 1 V03-001 MLJ0109 Martin L. Jack, 14-Apr-1983 12:48
47 0047 1 Changes for job controller baselevel.
48 0048 1
49 0049 1 **
```

```
51 0050 1 REQUIRE 'SRC$:JOBCTLDEF';
52 1091 1
53 1092 1
54 1093 1 FORWARD ROUTINE
55 1094 1 INITIALIZE_RESTRICTED: NOVALUE,
56 1095 1 HOURLY_AST_KERNEL, NOVALUE;
57 1096 1 HOURLY_AST:
58 1097 1
59 1098 1
60 1099 1 FORWARD
61 1100 1 LOCK_IPL;
62 1101 1
63 1102 1
64 1103 1 BIND
65 1104 1 TICKS_PER_HOUR= UPLIT(%X'61C46800', %X'00000008');
66 1105 1
67 1106 1
68 1107 1 LINKAGE
69 1108 1 SCH$FORCEDEXIT LINKAGE =
70 1109 1 JSB(REGISTER=0, REGISTER=3, REGISTER=4):
71 1110 1 NOPRESERVE(1, 2) PRESERVE(5) NOTUSED(6, 7, 8, 9, 10, 11);
72 1111 1
73 1112 1
74 1113 1 EXTERNAL ROUTINE
75 1114 1 SCH$FORCEDEXIT: SCH$FORCEDEXIT_LINKAGE ADDRESSING_MODE(GENERAL);
76 1115 1
77 1116 1
78 1117 1 EXTERNAL
79 1118 1 EXESGL_FLAGS: BITVECTOR ADDRESSING_MODE(GENERAL),
80 1119 1 SCH$GL_MAXPIX: ADDRESSING_MODE(GENERAL),
81 1120 1 SCH$GL_NULLPCB: ADDRESSING_MODE(GENERAL),
82 1121 1 SCH$GL_PCBVEC: REF VECTOR ADDRESSING_MODE(GENERAL);
83 1122 1
84 1123 1
85 1124 1 EXTERNAL LITERAL
86 1125 1 EXESV_EXPLICITP: UNSIGNED(6),
87 1126 1 EXESV_EXPLICITIS: UNSIGNED(6);
88 1127 1
89 1128 1
90 1129 1 BUILTIN
91 1130 1 ADDM,
92 1131 1 ASHQ,
93 1132 1 EDIV,
94 1133 1 EMUL,
95 1134 1 MTPR;
```



```

97 1135 1 GLOBAL ROUTINE INITIALIZE_RESTRICTED: NOVALUE=
98 1136 1
99 1137 1 ++
100 1138 1
101 1139 1 FUNCTIONAL DESCRIPTION:
102 1140 1 This routine initializes the restricted login hours subsystem.
103 1141 1
104 1142 1 INPUT PARAMETERS:
105 1143 1 NONE
106 1144 1
107 1145 1 IMPLICIT INPUTS:
108 1146 1 CUR_TIME - Current time.
109 1147 1
110 1148 1 OUTPUT PARAMETERS:
111 1149 1 NONE
112 1150 1
113 1151 1 IMPLICIT OUTPUTS:
114 1152 1 NONE
115 1153 1
116 1154 1 ROUTINE VALUE:
117 1155 1 NONE
118 1156 1
119 1157 1 SIDE EFFECTS:
120 1158 1 Next hourly timer set.
121 1159 1
122 1160 1 --
123 1161 1
124 1162 2 BEGIN
125 1163 2 LOCAL
126 1164 2 TIME_TEMP: VECTOR[2], ! Temporary for hourly calculation
127 1165 2 STATUS; ! Status return
128 1166 2
129 1167 2
130 1168 2 ! Compute the expiration time of the first hourly AST, and set the timer.
131 1169 2
132 1170 2 ASHQ(%REF(-5), CUR_TIME, TIME_TEMP);
133 1171 2 EDIV(%REF(%X'430E2340'), TIME_TEMP, TIME_TEMP[0], TIME_TEMP[1]);
134 1172 2 EMUL(%REF(%X'430E2340'), TIME_TEMP[0], %REF(0), TIME_TEMP);
135 1173 2 ASHQ(%REF(5), TIME_TEMP, HOURLY_TIME);
136 1174 2 ADDM(2, TICKS_PER_HOUR, HOURLY_TIME, HOURLY_TIME);
137 1175 2 HOURLY_PARAMS[0] = 4;
138 1176 2 STATUS = $SETIMR(
P 1177 2 DAYTIM=HOURLY_TIME,
P 1178 2 ASTADR=HOURLY_AST,
1179 2 REQIDT=JBC$K HOURLY IDT);
140 1180 2 IF NOT .STATUS THEN SIGNAL(JBC$_SETIMR OR STS$K_ERROR, 0, .STATUS);
141 1181 1 END;
142
143
```

.TITLE RESTRICT Restricted login hours enforcement
.IDENT \V04-000\

.PSECT COMMON,NOEXE, OVR,2

00000 DIAG_STORAGE BASE:
.BLKB 0
00000 DIAG_TRACE:

RESTRICT
V04-000

Restricted login hours enforcement

L 5
16-Sep-1984 00:24:02
14-Sep-1984 12:37:13

VAX-11 Bliss-32 V4.0-742
[JOBCTL.SRC]RESTRICT.B32;1

Page 4
(3)

	.BLKB	96
00060	DIAG_COUNT:	
	.BLKB	96
000C0	DIAG_FLAGS:	
	.BLKB	4
000C4	WORK_AREA:	
	.BLKB	44
000F0	SNDJBC_COUNT:	
	.BLKB	132
00174	GETQUI_COUNT:	
	.BLKB	40
0019C	SNDACC_COUNT:	
	.BLKB	28
001B8	SNDSMB_COUNT:	
	.BLKB	72
00200	DIAG_STORAGE_END:	
	.BLKB	0
00200	FLAGS:	
	.BLKB	4
00204	IMAGE_DUMP_STSFLG:	
	.BLKB	4
00208	THIS_SYSID:	
	.BLKB	6
0020E		
	.BLKB	2
00210	CUR_TIME:	
	.BLKB	8
00218	HOURLY_TIME:	
	.BLKB	8
00220	HOURLY_PARAMS:	
	.BLKB	20
00234	SYMBIONT_COUNT:	
	.BLKB	4
00238	QUEUE_REFERENCE_COUNT:	
	.BLKB	4
0023C	MBX_MESSAGE_COUNT:	
	.BLKB	4
00240	MBX:	
	.BLKB	4
00244	MBX_END:	
	.BLKB	4
00248	MEMORY_FREE_QUEUES:	
	.BLKB	40
00270	NONAST_WORK_QUEUE:	
	.BLKB	8
00278	BCB_FREE_LIST:	
	.BLKB	4
0027C	BCB_ACTIVE_LIST:	
	.BLKB	4
00280	GQL_FREE_LIST:	
	.BLKB	4
00284	GQL_ACTIVE_LIST:	
	.BLKB	4
00288	OPEN_GETQUI_LIST:	
	.BLKB	4
0028C	PROCESS_DATA_LIST:	
	.BLKB	4
00290	SYMBIONT_CONTROL:	
	.BLKB	4
00294	SPARE_AREA:	
	.BLKB	12

Restricted login hours enforcement

M 5
16-Sep-1984 00:24:02
14-Sep-1984 12:37:13

VAX-11 Bliss-32 V4.0-742
[JOBCTL.SRC]RESTRICT.B32;1

Page 5
(3)

002A0	REMOTE_REQUEST_LKSB:	
	.BLKB	8
002A8	QUEUE_FILE_LKSB:	
	.BLKB	8
002B0	QUEUE_LOCK_LKSB:	
	.BLKB	8
002B8	RSP:	.BLKB 8
002C0	JBC_PRIORITY:	
	.BLKB	4
002C4	JBC_PRIVILEGES:	
	.BLKB	8
002CC	JBC_QUOTAS:	
	.BLKB	66
0030E		.BLKB 2
00310	JBC_UIC:	.BLKB 4
00314	QUEUE_FAB:	
	.BLKB	80
00364	QUEUE_RAB:	
	.BLKB	68
003A8	QUEUE_NAM:	
	.BLKB	96
00408	QUEUE_XAB:	
	.BLKB	88
00460	QUEUE_RSA:	
	.BLKB	255
0055F		.BLKB 1
00560	QUEUE_ALQ:	
	.BLKB	4
00564	QUEUE_MBF:	
	.BLKB	1
00565		.BLKB 3
00568	ACCOUNTING_FABS:	
	.BLKB	8
00570	ACCOUNTING_RABS:	
	.BLKB	8
00578	ACCOUNT_FAB A:	
	.BLRB	80
005C8	ACCOUNT_RAB A:	
	.BLRB	68
0060C	ACCOUNT_NAM A:	
	.BLRB	96
0066C	ACCOUNT_RSA A:	
	.BLRB	255
0076B		.BLKB 1
0076C	ACCOUNT_FAB B:	
	.BLRB	80
007BC	ACCOUNT_RAB B:	
	.BLRB	68
00800	ACCOUNT_NAM B:	
	.BLRB	96
00860	ACCOUNT_RSA B:	
	.BLRB	255
0095F		.BLKB 1
00960	DIAG_FAB:	
	.BLKB	80
009B0	DIAG_RAB:	
	.BLKB	68


```

009F4 MBX_CHAN:
      .BLKB 4
009F8 MBX_IOSB:
      .BLKB 8
00A00 MBX_BUFFER:
      .BLKB 1024
00E00 VALUE_STORAGE_BASE:
      .BLKB 0
00E00 ITEM_PRESENT:
      .BLKB 32
00E20 VALUE_GETQUI_BASE:
      .BLKB 0
00E20 VALUE_ACCOUNTING_MESSAGE:
      .BLKB 8
00E26 VALUE_ACCOUNTING_TYPES:
      .BLKB 4
00E2A VALUE_AFTER_TIME:
      .BLKB 8
00E32 VALUE_ALIGNMENT_PAGES:
      .BLKB 1
00E33 VALUE_BASE_PRIORITY:
      .BLKB 1
00E34 VALUE_BATCH_INPUT:
      .BLKB 6
00E3A VALUE_BATCH_OUTPUT:
      .BLKB 10
00E44 VALUE_BUFFER_COUNT:
      .BLKB 1
00E45 VALUE_CHARACTERISTIC_NAME:
      .BLKB 6
00E4B VALUE_CHARACTERISTIC_NUMBER:
      .BLKB 1
00E4C VALUE_CHARACTERISTICS:
      .BLKB 16
00E5C VALUE_CHECKPOINT_DATA:
      .BLKB 8
00E62 VALUE_CLI:
      .BLKB 6
00E68 VALUE_CPU_DEFAULT:
      .BLKB 4
00E6C VALUE_CPU_LIMIT:
      .BLKB 4
00E70 VALUE_DESTINATION_QUEUE:
      .BLKB 8
00E78 VALUE_DEVICE_NAME:
      .BLKB 6
00E7E VALUE_ENTRY_NUMBER:
      .BLKB 4
00E82 VALUE_ENTRY_NUMBER_OUTPUT:
      .BLKB 10
00E8C VALUE_EXTEND_QUANTITY:
      .BLKB 2
00E8E VALUE_FILE_COPIES:
      .BLKB 1
00E8F VALUE_FILE_IDENTIFICATION:
      .BLKB 36
00EB3 VALUE_FILE_SETUP_MODULES:

```

```

00EB9 VALUE_FILE SPECIFICATION:
      .BLKB 6
00EBF VALUE_FIRST PAGE:
      .BLKB 6
00EC3 VALUE_FORM DESCRIPTION:
      .BLKB 4
00EC9 VALUE_FORM LENGTH:
      .BLKB 6
00ECA VALUE_FORM MARGIN_BOTTOM:
      .BLKB 1
00ECB VALUE_FORM MARGIN_LEFT:
      .BLKB 2
00ECD VALUE_FORM MARGIN_RIGHT:
      .BLKB 2
00ECF VALUE_FORM MARGIN_TOP:
      .BLKB 1
00ED0 VALUE_FORM NAME:
      .BLKB 6
00ED6 VALUE_FORM NUMBER:
      .BLKB 4
00EDA VALUE_FORM:
      .BLKB 8
00EE2 VALUE_FORM SETUP MODULES:
      .BLKB 6
00EE8 VALUE_FORM STOCK:
      .BLKB 6
00EEE VALUE_FORM WIDTH:
      .BLKB 2
00EF0 VALUE_GENERIC_TARGET:
      .BLKB 996
012D4 VALUE_JOB COPIES:
      .BLKB 1
012D5 VALUE_JOB LIMIT:
      .BLKB 1
012D6 VALUE_JOB NAME:
      .BLKB 6
012DC VALUE_JOB RESET_MODULES:
      .BLKB 6
012E2 VALUE_JOB SIZE_MAXIMUM:
      .BLKB 4
012E6 VALUE_JOB SIZE_MINIMUM:
      .BLKB 4
012EA VALUE_JOB STATUS_OUTPUT:
      .BLKB TO
012F4 VALUE_LAST PAGE:
      .BLKB 4
012F8 VALUE_LIBRARY SPECIFICATION:
      .BLKB 6
012FE VALUE_LOG QUEUE:
      .BLKB 8
01306 VALUE_LOG SPECIFICATION:
      .BLKB 6
0130C VALUE_NOTE:
      .BLKB 6
01312 VALUE_OPERATOR_REQUEST:
      .BLKB 6

```



```

01318 VALUE_OWNER UIC:
      .BLKB 4
0131C VALUE_PAGE_SETUP MODULES:
      .BLKB 6
01322 VALUE_PARAMETER_1:
      .BLKB 6
01328 VALUE_PARAMETER_2:
      .BLKB 6
0132E VALUE_PARAMETER_3:
      .BLKB 6
01334 VALUE_PARAMETER_4:
      .BLKB 6
0133A VALUE_PARAMETER_5:
      .BLKB 6
01340 VALUE_PARAMETER_6:
      .BLKB 6
01346 VALUE_PARAMETER_7:
      .BLKB 6
0134C VALUE_PARAMETER_8:
      .BLKB 6
01352 VALUE_PRIORITY:
      .BLKB 1
01353 VALUE_PROCESSOR:
      .BLKB 6
01359 VALUE_PROTECTION:
      .BLKB 4
0135D VALUE_QUEUE:
      .BLKB 6
01363 VALUE_QUEUE_FILE SPECIFICATION:
      .BLKB 6
01369 VALUE_RELATIVE_PAGE:
      .BLKB 4
0136D VALUE_RESERVED_INPUT_1:
      .BLKB 1
0136E VALUE_RESERVED_INPUT_2:
      .BLKB 2
01370 VALUE_RESERVED_INPUT_3:
      .BLKB 4
01374 VALUE_RESERVED_INPUT_4:
      .BLKB 6
0137A VALUE_RESERVED_OUTPUT_1:
      .BLKB 10
01384 VALUE_RESERVED_OUTPUT_2:
      .BLKB 10
0138E VALUE_SEARCH_STRING:
      .BLKB 6
01394 VALUE_SC$NODE_NAME:
      .BLKB 6
0139A VALUE_WSDEFAULT:
      .BLKB 2
0139C VALUE_W$EXTENT:
      .BLKB 2
0139E VALUE_W$QUOTA:
      .BLKB 2
013A0 VALUE_STORAGE_END:
      .BLKB 0

```


.PSECT CODE, NOWRT, 2

00000008 61C46800 00000 P.AAA: .LONG 1640261632, 8

JBC\$_CLOSEOUT= 266328
JBC\$_NOCMKRNL= 272388
JBC\$_NOOPER= 272532
JBC\$_NOSYSNAM= 272404
JBC\$_OPENIN= 266392
JBC\$_OPENOUT= 266400
JBC\$_READERR= 266416
JBC\$_WRITEERR= 266448
TICKS_PER_HOUR= P.AAA.EXTRN SCH\$FORCEDEXIT, EXES\$GL_FLAGS
.EXTRN SCH\$GL_MAXPIX, SCH\$GL_NULLPCB
.EXTRN SCH\$GL_PCBVEC, EXESV_EXPLICITP
.EXTRN EXESV_EXPLICITP
.EXTRN SYS\$SETIMR

04	AE	7E	F8	52	00000000'	EF	9E	00002	.ENTRY	INITIALIZE RESTRICTED, Save R2	:	1135
		6E		A2	FB	8F	79	00009	MOVAB	HOURLY_TIME, R2	:	
				6E	430E2340	8F	7B	0000F	ASHQ	#-5, CUR TIME, TIME_TEMP	:	1170
									EDIV	#1125000000, TIME_TEMP, TIME_TEMP, -	:	1171
										TIME_TEMP+4	:	
				6E	430E2340	8F	7A	00019	EMUL	#1125000000, TIME_TEMP, #0, TIME_TEMP	:	1172
				6E		05	79	00022	ASHQ	#5, TIME_TEMP, HOURLY_TIME	:	1173
				62	CF	AF	C0	00026	ADDL2	TICKS_PER_HOUR, HOURLY_TIME	:	1174
			04	A2	CF	AF	D8	0002A	ADWC	TICKS_PER_HOUR+4, HOURLY_TIME+4	:	
			08	A2		04	D0	0002F	MOVL	#4, HOURLY_PARAMS	:	1175
						02	DD	00033	PUSHL	#2	:	1179
					0000V	CF	9F	00035	PUSHAB	HOURLY_AST	:	
						52	DD	00039	PUSHL	R2	:	
						7E	D4	0003B	CLRL	-(SP)	:	
				00000000G	00	04	FB	0003D	CALLS	#4, SYS\$SETIMR	:	
					11	50	E8	00044	BLBS	STATUS, 1\$	:	1180
						50	DD	00047	PUSHL	STATUS	:	
						7E	D4	00049	CLRL	-(SP)	:	
					0004845A	8F	DD	0004B	PUSHL	#296026	:	
				00000000G	00	03	FB	00051	CALLS	#3, LIB\$SIGNAL	:	
						04	00058	1\$:	RET		:	1181

; Routine Size: 89 bytes, Routine Base: CODE + 0008

```
1182 1 ROUTINE HOURLY_AST_KERNEL(PIX, HOUR, DAY, MODE)=
1183 1
1184 1 ++
1185 1
1186 1 FUNCTIONAL DESCRIPTION:
1187 1 This routine tests to see if a process is restricted from logins at the
1188 1 current time, and if so, queues an AST to the process to cause its
1189 1 deletion. This routine executes in kernel mode and partially at
1190 1 IPL$_SYNCH.
1191 1
1192 1 INPUT PARAMETERS:
1193 1 PIX - Process index.
1194 1 HOUR - Current hour of day (0-23).
1195 1 DAY - Current day of week (0-6).
1196 1 MODE - Access mode at which to queue AST.
1197 1
1198 1 IMPLICIT INPUTS:
1199 1 PCB for the specified process.
1200 1 EXESGL_FLAGS
1201 1
1202 1 OUTPUT PARAMETERS:
1203 1 NONE
1204 1
1205 1 IMPLICIT OUTPUTS:
1206 1 NONE
1207 1
1208 1 ROUTINE VALUE:
1209 1 True if AST queued to process, false otherwise.
1210 1
1211 1 SIDE EFFECTS:
1212 1 AST may be queued to process.
1213 1
1214 1 --
1215 1
1216 2 BEGIN
1217 2 REGISTER
1218 2 RPIX, ! Copy of PIX
1219 2 RHOURL, ! Copy of HOUR
1220 2 RDAY, ! Copy of DAY
1221 2 PCB: REF BBLOCK, ! Pointer to PCB
1222 2 JIB: REF BBLOCK, ! Pointer to JIB
1223 2 STATUS; ! Routine status
1224 2 STACKLOCAL
1225 2 SMODE; ! Copy of MODE
1226 2
1227 2
1228 2 ! Load parameters into registers or onto the kernel stack to avoid page fault
1229 2 ! at elevated IPL.
1230 2
1231 2 RPIX = .PIX;
1232 2 RHOURL = .HOUR;
1233 2 RDAY = .DAY;
1234 2 SMODE = .MODE;
1235 2
1236 2
1237 2 ! If this is the first execution of this routine after midnight, clear the
1238 2 ! operator override flag.
```



```
202 1239 2 !
203 1240 2 IF .RHOURL EQL 0 AND .SMODE EQL PSL$C_USER
204 1241 2 THEN
205 1242 2 EXE$GL_FLAGS[EXE$V_EXPLICIT] = FALSE;
206 1243 2
207 1244 2
208 1245 2 ! Raise IPL to SYNCH.
209 1246 2
210 1247 2 MTPR(LOCK_IPL, PR$_IPL);
211 1248 2
212 1249 2
213 1250 2 ! Do the real work.
214 1251 2
215 1252 2 PCB = .SCH$GL_PCBVEC[.RPIX];
216 1253 2 IF
217 1254 2 BEGIN
218 1255 2 IF .PCB EQLA SCH$GL_NULLPCB ! Null process: hands off
219 1256 2 THEN
220 1257 2 FALSE
221 1258 2 ELSE
222 1259 2 IF .PCB[PCB$L_OWNER] NEQ 0 ! Subprocess: likewise
223 1260 2 OR .PCB[PCB$V_NETWRK] ! Network process: likewise
224 1261 2 THEN
225 1262 2 FALSE
226 1263 2 ELSE
227 1264 2 BEGIN
228 1265 2 JIB = .PCB[PCB$L_JIB]; ! Point to JIB
229 1266 2 IF .JIB EQL 0 ! No JIB
230 1267 2 THEN
231 1268 2 FALSE
232 1269 2 ELSE
233 1270 2 IF
234 1271 2 BEGIN
235 1272 2 IF .EXE$GL_FLAGS[EXE$V_EXPLICIT] ! Block false if primary day
236 1273 2 THEN ! Operator override?
237 1274 2 .EXE$GL_FLAGS[EXE$V_EXPLICIT] ! Clear if operator override primary day
238 1275 2 ELSE
239 1276 2 .BITVECTOR[JIB[JIB$L_DAYTYPES], .RDAY] ! Clear if process specified primary day
240 1277 2 END
241 1278 2 THEN
242 1279 2 .BITVECTOR[JIB[JIB$L_ODAYHOURS], .RHOURL] ! Set if restricted secondary hour
243 1280 2 ELSE
244 1281 2 .BITVECTOR[JIB[JIB$L_PDAYHOURS], .RHOURL] ! Set if restricted primary hour
245 1282 2 END
246 1283 2 END
247 1284 2 THEN
248 1285 2 BEGIN
249 1286 2 SCH$FORCEDEXIT(SMODE, SSS$_FORCEDEXIT, .PCB); ! Queue AST to process
250 1287 2 STATUS = TRUE;
251 1288 2 END
252 1289 2 ELSE
253 1290 2 STATUS = FALSE;
254 1291 2
255 1292 2
256 1293 2 ! Lower IPL, and return status.
257 1294 2
258 1295 2 MTPR(%REF(0), PR$_IPL);
```

RESTRICT
V04-000

Restricted login hours enforcement

G 6
16-Sep-1984 00:24:02
14-Sep-1984 12:37:13

VAX-11 Bliss-32 V4.0-742
[JOBCTL.SRC]RESTRICT.B32;1

Page 12
(4)

: 259
: 260

1296 2 .STATUS
1297 1 END;

```
003C 00000 HOURLY_AST KERNEL:
      .WORD Save R2,R3,R4,R5
55 00000000G 00 9E 00002 MOVAB EXESGL_FLAGS, R5
50 04 AC 7D 00009 MOVQ PIX, RPIX
52 0C AC D0 0000D MOVL DAY, RDAY
      10 AC DD 00011 PUSHL MODE
      51 D5 00014 TSTL RHOURL
      09 12 00016 BNEQ 1$
03 6E D1 00018 CMPL SMODE, #3
      04 12 0001B BNEQ 1$
00 65 00G E5 0001D BBCC S*EXESV_EXPLICIT, EXESGL_FLAGS, 1$
      12 0000V CF DA 00021 1$: MTPR LOCK_IPC, #18
53 00000000G 00 D0 00026 MOVL SCH$GL_PCBVEC, R3
54 6340 D0 0002D MOVL (R3)[RPIX], PCB
50 00000000G 00 9E 00031 MOVAB SCH$GL_NULLPCB, R0
50 54 D1 00038 CMPL PCB, R0
      3F 13 0003B BEQL 6$
      1C A4 D5 0003D TSTL 28(PCB)
      3A 12 00040 BNEQ 6$
35 26 A4 05 E0 00042 BBS #5, 38(PCB), 6$
50 0080 C4 D0 00047 MOVL 128(PCB), JIB
      2E 13 0004C BEQL 6$
06 65 00G E1 0004E BBC S*EXESV_EXPLICIT, EXESGL_FLAGS, 2$
0E 65 00G E1 00052 BBC S*EXESV_EXPLICITP, EXESGL_FLAGS, 4$
      05 11 00056 BRB 3$
07 08 A0 52 E1 00058 2$: BBC RDAY, 11(JIB), 4$
1A 64 A0 51 E1 0005D 3$: BBC RHOURL, 100(JIB), 6$
      05 11 00062 BRB 5$
13 60 A0 51 E1 00064 4$: BBC RHOURL, 96(JIB), 6$
50 6E 9E 00069 5$: MOVAB SMODE, R0
53 217C 8F 3C 0006C MOVZWL #8572, R3
      00 16 00071 JSB SCH$FORCEDEXIT
50 00000000G 01 D0 00077 MOVL #1, STATUS
      02 11 0007A BRB 7$
      50 D4 0007C 6$: CLRL STATUS
12 00 DA 0007E 7$: MTPR #0, #18
      04 00081 RET
```

; Routine Size: 130 bytes, Routine Base: CODE + 0061

RESTRICT
V04-000

Restricted login hours enforcement

H 6
16-Sep-1984 00:24:02
14-Sep-1984 12:37:13

VAX-11 Bliss-32 V4.0-742
[JOBCTL.SRC]RESTRICT.B32;1

Page 13
(5)

```
: 262      1298 1 DWN
: 263      1299 1
: 264      1300 1      LOCK_IPL:      PSECT(CODE) INITIAL(IPL$_SYNCH);
: 265      1301 1
: 266      L 1302 1 %IF LOCK_IPL + 4 - HOURLY_AST_KERNEL GTR 512
: 267      1303 1 %THEN %ERROR('Possible page fault at elevated IPL') %FI
```

```
269 1304 1 ROUTINE HOURLY_AST(TYPE): NOVALUE=
270 1305 1
271 1306 1 ++
272 1307 1
273 1308 1 FUNCTIONAL DESCRIPTION:
274 1309 1 This is the completion AST routine for the timer that expires on the
275 1310 1 hour. It checks for processes that are restricted by the authorization
276 1311 1 record from logins at the current hour, queues ASTs to these processes
277 1312 1 to cause their deletion, and sets a new timer. If any processes were
278 1313 1 found, it sets a timer to expire in 2 minutes to check again; otherwise
279 1314 1 it sets a new hourly timer.
280 1315 1
281 1316 1 INPUT PARAMETERS:
282 1317 1 TYPE - JBC$K_HOURLY_IDT or JBC$K_MINUTE_IDT,
283 1318 1 passed as REQIDT.
284 1319 1
285 1320 1 IMPLICIT INPUTS:
286 1321 1 HOURLY_TIME - Expiration time of this AST.
287 1322 1
288 1323 1 OUTPUT PARAMETERS:
289 1324 1 NONE
290 1325 1
291 1326 1 IMPLICIT OUTPUTS:
292 1327 1 HOURLY_TIME - Expiration time of next AST.
293 1328 1
294 1329 1 ROUTINE VALUE:
295 1330 1 NONE
296 1331 1
297 1332 1 SIDE EFFECTS:
298 1333 1 ASTs queued to processes that should not be logged in at this hour.
299 1334 1
300 1335 1 --
301 1336 1
302 1337 2 BEGIN
303 1338 2 LOCAL
304 1339 2 POSSIBLE,
305 1340 2 STATUS;
306 1341 2
307 1342 2
308 1343 2 IF .TYPE EQL JBC$K_HOURLY_IDT
309 1344 2 THEN
310 1345 2 BEGIN
311 1346 2 LOCAL
312 1347 2 TIME_TEMP: VECTOR[2], ! Temporary for time calculations
313 1348 2 T; ! Temporary for time calculations
314 1349 2
315 1350 2
316 1351 2 ! Get the current time.
317 1352 2
318 1353 2 $GETTIM(TIMADR=CUR_TIME);
319 1354 2
320 1355 2
321 1356 2 ! Compute the day of the week and hour of the day, and initialize the AST
322 1357 2 ! access mode to user mode.
323 1358 2
324 1359 2 EDIV(%REF(600000000), CUR_TIME, TIME_TEMP[0], T); ! Get minutes since base time value
325 1360 2 TIME_TEMP[1] = 0;
```



```

326 1361 3 EDIV(%REF(60), TIME_TEMP, TIME_TEMP[0], T); ! Get hours since base time value
327 1362 3 TIME_TEMP[1] = 0;
328 1363 3 EDIV(%REF(24), TIME_TEMP, TIME_TEMP[0], HOURLY_PARAMS[2]); ! Get days since base time value, store hour
329 1364 3 TIME_TEMP[0] = TIME_TEMP[0] + 2; ! Adjust for base day being Wednesday
330 1365 3 TIME_TEMP[1] = 0;
331 1366 3 EDIV(%REF(7), TIME_TEMP, TIME_TEMP[0], HOURLY_PARAMS[3]); ! Get day of week
332 1367 3 HOURLY_PARAMS[4] = PSL$C_USER;
333 1368 3
334 1369 3
335 1370 3 ! Compute the expiration time of the next hourly AST, and set the timer.
336 1371 3
337 1372 3 ASHQ(%REF(-5), CUR TIME, TIME_TEMP);
338 1373 3 EDIV(%REF(%X'430E2340'), TIME_TEMP, TIME_TEMP[0], T);
339 1374 3 EMUL(%REF(%X'430E2340'), TIME_TEMP[0], %REF(0), TIME_TEMP);
340 1375 3 ASHQ(%REF(5), TIME_TEMP, HOURLY_TIME);
341 1376 3 ADDM(2, TICKS_PER_HOUR, HOURLY_TIME, HOURLY_TIME);
342 1377 3 END;
343 1378 3
344 1379 3
345 1380 3 ! Scan all processes looking for processes to delete, and delete these.
346 1381 3
347 1382 3 POSSIBLE = FALSE;
348 1383 3 INCR PIX FROM 2 TO .SCH$GL_MAXPIX DO
349 1384 3 BEGIN
350 1385 3 HOURLY_PARAMS[1] = .PIX;
351 1386 3 POSSIBLE = .POSSIBLE OR $CMKRN(LROUTIN=HOURLY_AST_KERNEL, ARGST=HOURLY_PARAMS);
352 1387 3 END;
353 1388 3
354 1389 3
355 1390 3 ! If any processes were found and the current AST access mode is not kernel,
356 1391 3 ! set a timer to expire in 2 minutes so that we may check again and queue ASTs
357 1392 3 ! at the next outer access mode. Otherwise, set a new hourly timer.
358 1393 3
359 1394 3 IF
360 1395 3 BEGIN
361 1396 3 IF .POSSIBLE
362 1397 3 THEN
363 1398 3 BEGIN
364 1399 3 HOURLY_PARAMS[5] = .HOURLY_PARAMS[5] - 1;
365 1400 3 .HOURLY_PARAMS[5] GEQ 0
366 1401 3 END
367 1402 3 ELSE
368 1403 3 FALSE
369 1404 3 END
370 1405 3 THEN
371 1406 3 STATUS = $SETIMR(
372 1407 3 DAYTIM=UPLIT(-1200000000, -1), ! 2 minute delta time
373 1408 3 ASTADR=HOURLY_AST,
374 1409 3 REQIDT=JBC$K_MINUTE_IDT)
375 1410 3 ELSE
376 1411 3 STATUS = $SETIMR(
377 1412 3 DAYTIM=HOURLY_TIME,
378 1413 3 ASTADR=HOURLY_AST,
379 1414 3 REQIDT=JBC$K_HOURLY_IDT);
380 1415 3
381 1416 3 IF NOT .STATUS THEN SIGNAL(JBC$_SETIMR OR STS$K_ERROR, 0, .STATUS);
382 1417 3 END;
```

```
00000008 000E3 .BLKB 1
000E4 LOCK_IPL:
FFFFFFFF B8797400 000E8 P.AAB: .LONG 8
                                           .LONG -1200000000, -1
                                           .EXTRN SYSS$GETTIM, SYSS$CMKRNL

007C 00000 HOURLY_AST:
56 FFOA CF 9E 00002 .WORD Save R2,R3,R4,R5,R6 1304
55 00000000 EF 9E 00007 MOVAB TICKS_PER_HOUR, R6
5E 04 08 C2 0000E MOVAB HOURLY_TIME, R5
02 04 AC D1 00011 SUBL2 #8, SP
                                CMPL TYPE, #2 1343
                                BNEQ 1$
                                PUSHAB CUR_TIME 1353
00 00000000G 00 04 01 FB 0001A CALLS #1, SYSS$GETTIM
6E F8 A5 23C34600 8F 7B 00021 EDIV #600000000, CUR_TIME, TIME_TEMP, T 1359
                                CLRL TIME_TEMP+4 1360
50 6E 6E 04 3C 7B 0002E EDIV #60, TIME_TEMP, TIME_TEMP, T 1361
50 6E 6E 04 AE D4 00033 CLRL TIME_TEMP+4 1362
10 A5 6E 6E 18 7B 00036 EDIV #24, TIME_TEMP, TIME_TEMP, HOURLY_PARAMS+8 1363
6E 6E 02 C0 0003C ADDL2 #2, TIME_TEMP 1364
                                CLRL TIME_TEMP+4 1365
14 A5 6E 6E 07 7B 00042 EDIV #7, TIME_TEMP, TIME_TEMP, HOURLY_PARAMS+12 1366
                                MOVL #3, HOURLY_PARAMS+18 1367
6E 18 A5 03 D0 00048 ASHQ #5, CUR_TIME, TIME_TEMP 1372
6E F8 A5 FB 8F 79 0004C EDIV #1125000000, TIME_TEMP, TIME_TEMP, T 1373
6E 6E 6E 430E2340 8F 7B 00052 EMUL #1125000000, TIME_TEMP, #0, TIME_TEMP 1374
6E 6E 6E 430E2340 8F 7A 0005B ASHQ #5, TIME_TEMP, HOURLY_TIME 1375
65 65 66 C0 00068 ADDL2 TICKS_PER_HOUR, HOURLY_TIME 1376
                                ADWC TICKS_PER_HOUR+4, HOURLY_TIME+4
                                MOVL SCH$GC_MAXPIX, R4 1383
04 04 A5 04 A6 D8 0006B BRB 3$
54 00000000G 00 D0 00070 1$: MOVQ #1, PIX
52 52 15 11 0007A BRB 3$
                                MOVL PIX, HOURLY_PARAMS+4 1385
                                PUSHAB HOURLY_PARAMS 1386
                                PUSHAB HOURLY_AST KERNEL
                                CALLS #2, SYSS$CMKRNL
                                BISL2 R0, POSSIBLE
                                AOBLEQ R4, PIX, 2$ 1383
                                BLBC POSSIBLE, 4$ 1396
                                DECL HOURLY_PARAMS+20 1399
                                BLSS 4$ 1400
                                PUSHL #3 1409
                                PUSHAB HOURLY_AST
                                PUSHAB P.AAB
                                BRB 5$
                                PUSHL #2 1414
                                PUSHAB HOURLY_AST
                                PUSHL R5
                                CLRL -(SP)
                                CALLS #4, SYSS$SETIMR 1416
                                BLBS STATUS, 6$
                                PUSHL STATUS
```


RESTRICT
V04-000

Restricted login hours enforcement

L 6
16-Sep-1984 00:24:02
14-Sep-1984 12:37:13

VAX-11 Bliss-32 V4.0-742
[JOBCTL.SRC]RESTRICT.B32;1

Page 17
(6)

00000000G 00 0004845A 7E D4 000BF CLRL -(SP)
8F DD 000C1 PUSH #296026
03 FB 000C7 CALLS #3, LIB\$SIGNAL
04 000CE 6\$ RET

:
:
:
: 1417

; Routine Size: 207 bytes, Routine Base: CODE + 00F0

RESTRICT
V04-000

Restricted login hours enforcement

M 6
16-Sep-1984 00:24:02
14-Sep-1984 12:37:13

VAX-11 Bliss-32 V4.0-742
[JOBCTL.SRC]RESTRICT.B32;1

Page 18
(7)

: 384
: 385
1418 1 END
1419 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
COMMON	5024	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, OVR, NOPIC, ALIGN(2)
CODE	447	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	49	0	1000	00:01.5

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:RESTRICT/OBJ=OBJ\$:RESTRICT MSRC\$:RESTRICT/UPDATE=(ENH\$:RESTRICT)

: Size: 426 code + 5045 data bytes
: Run Time: 00:14.7
: Elapsed Time: 00:53.8
: Lines/CPU Min: 5799
: Lexemes/CPU-Min: 49520
: Memory Used: 237 pages
: Compilation Complete

0194 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RESTRICT
LIS

RESPONSE
LIS

SNOOZE
LIS

SCHEDULER
LIS